Interim Report to Goddard Space Flight Center on

TELE-AUTONOMOUS CONTROL INVOLVING CONTACTS

- The Applications of A High Precision Laser Line Range Sensor -

by
R. A. Volz
L. Shao
M.W. Walker
and
L.A. Conway

Contract Number NAG5-1024

University of Michigan
Electrical Engineering and Computer Science Dept.
The Robotics Research Laboratory
Ann Arbor, Michigan 48109-2110

Texas A&M University
Computer Science Department
College Station, Tesax 77843

May 22, 1989

(NASA-CR-184995) TELE-AUTONOMOUS CONTROL INVOLVING CONTACTS: THE APPLICATIONS OF A HIGH PRECISION LASER LINE RANGE SENSOR Interim Report (Texas A&M Univ.) 17 p CSCL 13B G3/31

N89-24509

Unclas 0210901

Tele-Autonomous Control Involving Contacts

– The Applications of A High Precision Laser Line Range Sensor –

1 Introduction

The difficulty in applying tele-autonomous control to do assembly or other similar tasks is the existence of uncertainties of the environment including the telerobot itself. To deal with this difficulty, the remote system must have a certain level of intelligence to either **adapt** or **avoid** these uncertainties. There has been some success with the **adapt** strategy such as force and compliance controlled assembly [25], path planning and replanning [26] [27], simulated force/torque reflection control [7] and *etc.*. But most of them are not quite flexible and have only very limited use in real applications. The **avoid** strategy is the most widely used in industrial systems. There are two methods of avoiding uncertainties in robot assembly: precision assembly techniques and localization and assembly. The precision assembly uses a set of fixtures or jigs to accurately control the position of the objects to be operated and thus avoiding the errors during the assembly. The problem with this method is its inflexibility and is very difficult to be used in space applications. The localization and assembly is therefore a good choice. We are not surprised that the "Locate" operation is defined as one of the basic building blocks of E-move in RATS [20]. Our last report has already given a detailed discussion about object localization techniques. They can be divided into *Recognition-Localization Techniques* and *Direct Localization Techniques*. Many early object localization algorithms such as [6] [10] [15] [23] concentrate on the recognition-localization techniques. technique. The researchers have not paid much attention on the direct localization technique until recently they found out that a lot of robot applications, especially in assembly and inspection tasks, object localization algorithms with high accuracy and real-time execution are the key factors to succeed these tasks. For example, in RATS [20], the "Locate" operation should be finished in the range of seconds. Those early algorithms are not suitable for this situation.

Recently, several algorithms which are directed specially at accurate and fast location determination have been proposed. One algorithm presented by Gunnarsson and Prinz [8, 9] uses sparse range data measured on the surfaces of an object to locate the object. Their algorithm tries to find a transformation which will make the sum of distances between each measured point and its correspondent in transformed surface minimal. Gordon and Seering [12] have developed a system which uses striped-light and camera sensing system to locate an object. The image projected by a striped-light on the surfaces of an object is viewed by a camera and a line-surface matching is then used to carry out the transformation parameter estimation. The same striped-light and camera sensing system is also used by Rutkowski, Benton and *etc.* [1, 21]. But their matching strategy is point-surface matching. All of the algorithms can be implemented fast and accurately. But if

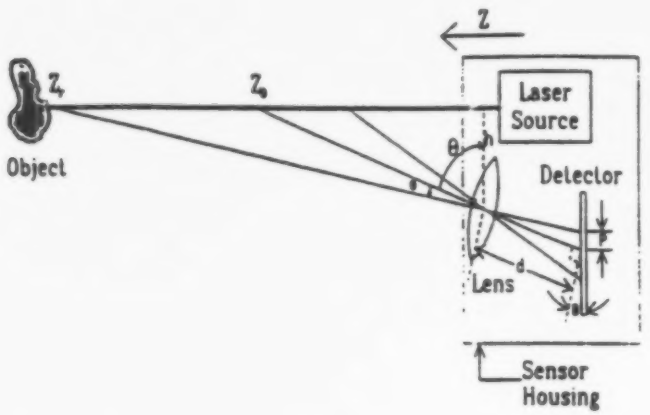

Figure 1: Geometry of a Laser Based Triangulation Sensor. ([14])

the object is a polygon, at least three surfaces need to be accessed. Also, the iterative numerical procedures used by some of the algorithms have convergence problems.

In this paper, we present an object localization algorithm which is based on line-line matching. A high precision laser line range sensor is used to make a sequence of measurements on the object and then to extract line features from these measured data. The line features could be boundary edges for planar surfaces or axes for surfaces of revolution. The line features are matched to corresponding modeled line features. Closed form formulas are used to carry out all the computation. The system described in this paper requires a highly constrained environment. That is, either the environment is a highly-structured or the position relationships among the objects in the environment have previously been established approximately. This is a reasonable assumption for many space applications. With this assumption, fairly fast and reliable measurements can be taken, which in turn will lead to a fast and accurate localization.

2 Sensing System

2.1 Sensor Requirements

There are a number of three-dimensional range sensing systems [17]. In our application, the sensing system should meet the following requirements:

- A high measurement rate: each complete measurement (including the time spent on feature extraction and location computation) should be finished within one second.
- High measurement accuracy.
- Capable of extracting line-segment features from sensed object.

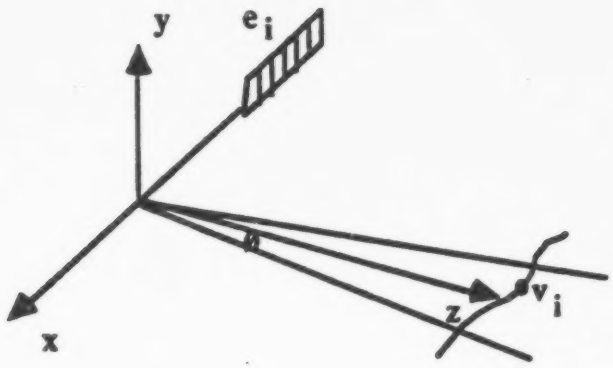

Figure 2: Line range sensor configuration

A line range sensor manufactured by CyberOptics Corp. was selected as a prototype sensor in our experiments because it satisfies with most of the requirements. This sensor is a laser triangulation based range sensor and has the following specifications: [5]

- Measurement rate: 5 frames/second.
- Standoff: 30 mm.
- Depth range: 4 mm.
- Line length: 2 mm.
- The number of pixels along the line: up to 64.
- Lateral resolution: 30 micrometer.
- Depth resolution: 10 micrometer.
- Field of View: 30 degree.

2.2 3-D Range Sensing Fundamentals

The triangulation based range finding technique uses a known geometric structure between the source of laser beam and the detectors to determine the distance. Figure 1 shows the geometry of such sensors making a spot measurement. Any position along the laser source beam or range can be determined from the position image on the detector by using the relationship:

$$Z_r = h \times \tan[\tan^{-1}(\frac{Z_0}{h}) + \tan^{-1}(\frac{p}{d\cos B})] \quad \text{(see[14])}$$

The line range sensor uses the same principle. If we assign a left-hand coordinate system to the sensor (see Fig.2), the coordinates of points of the object in 3-D space lying on the

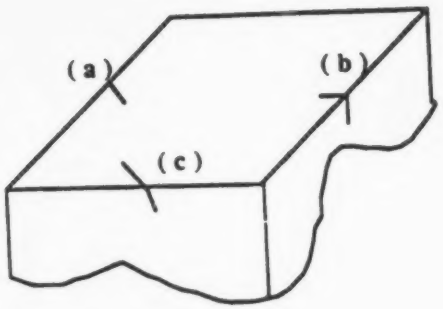

Figure 3: The boundary edge sensing patterns

projected line may be derived as a function of the readings of the sensor detector array. Letθ be the angle of filed of view, k the number of pixels of the detector array, and e_i the range reading of pixel i, where $1 \leq i \leq k$, then the coordinates of the point $\mathbf{v}_i$ in space corresponding to pixel i is

$$\mathbf{v}_i = \begin{bmatrix} \frac{1}{2}\tan\theta(\frac{2}{1-k}i - \frac{1+k}{1-k})e_i \\ 0 \\ e_i \end{bmatrix}$$

This information will be used in the next section as the input to extract line-segment parameters.

3 Feature Extraction and Location Determination

The feature extraction and localization process basically consists of two steps: the line-segment parameter extraction step and location determination step. Two cases are considered in the line-segment feature extraction step:

1. The line-segment to be extracted is a boundary edge of a planar surface;
2. The line-segment to be extracted is an axis of a surface of revolution.

In the following, we will give a brief description of the extraction process for each type of line-segment.

3.1 Boundary Edge Parameter Extraction

Any boundary edge of a planar surface can be considered as an intersection of two planes. One is the planar surface itself. Another plane is the one perpendicular to it. The process

of extracting boundary edge parameters is first to take a sequence of measurements using the line range sensor along the boundary edge of the planar surface, and then to determine parameters of the two mentioned planes from these measured data.

After each measurement, the following processing tasks are performed:

1. Detect boundary point: determine whether there exists any readings in the sensor detector array which is corresponding to a boundary point of the object surface and locate it if the answer is yes. The detection of boundary point is done by analyzing the patterns of readings from the detector array. One possible pattern of the readings is that a part of the cells of the detector array has no output while in other part of the cells the readings do exist. Another pattern is a change in slope in the readings. These patterns correspond to the cases (a) and (b) or (c) of Fig. 3 respectively.

2. Eliminate bad readings to reduce the inaccuracies due to noise and object distortion. The elimination is done by a histogram technique.

Having found the boundary point and eliminated the unreliable data from the measurements, we are ready to compute the needed line-segment parameters. First, all the remaining data measured on the planar surface are used to determine the parameters of the surface. A least squares fitting can be used. Suppose the equation of the planar surface to be determined is $ax + by + z = d$, we need to find the a, b and d from a set of k readings (x_i, y_i, z_i), where $i = 1, \cdots, k$ and $k > 3$. The following equation can be obtained and used for a least squares optimization and can be solved by a closed form formula:

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ & \vdots & \\ x_k & y_k & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ -d \end{bmatrix} = \begin{bmatrix} -z_1 \\ -z_2 \\ \vdots \\ -z_k \end{bmatrix}$$

Once the plane equation is obtained, the surface normal $\mathbf{n} = (n_x, n_y, n_z)$ can be calculated from it. Next, we need to find out the equation for the second plane. This is done by a projection process. That is, a rotation matrix $\mathbf{M}$ is made in such a way that the z axis in the coordinate system is rotated to the direction of the normal vector $\mathbf{n}$ of the plane, see Fig. 4. All the boundary points will then be projected on the plane by multiplying $\mathbf{M}$ with these points and taking only x and y components of the multiplication. These projected two dimensional boundary points are used to compute an optimal two dimensional line-segment. Back projection of the two dimensional line-segment will give the required plane equation. The rotation matrix has the property that it is an orthonormal matrix

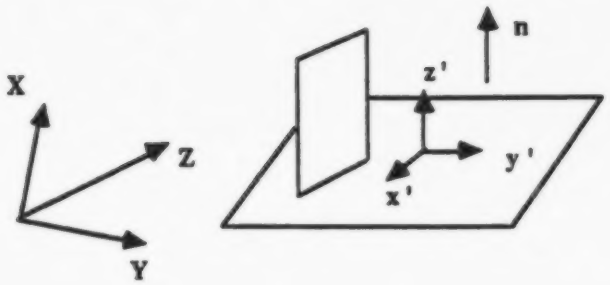

Figure 4: The use of projection method to find the equation of second plane

and that $\mathbf{Mn} = (0\ \ 0\ \ 1)^T$. The matrix is not unique. One of them has the form:

$$\mathbf{M} = \begin{bmatrix} \frac{n_y}{g} & -\frac{n_x}{g} & 0 \\ \frac{n_x n_z}{g} & \frac{-n_y n_z}{g} & -g \\ n_x & n_y & n_z \end{bmatrix}$$

where $g = \sqrt{n_x^2 + n_y^2}$.
Thus, the line-segment in the original coordinate system can be expressed as the following form:

$$\begin{cases} a_1 x + b_1 y + z = d_1 \\ a_2 x + b_2 y + z = d_2 \end{cases}$$

Other forms of representation can be obtained from it easily.

3.2 Axis Parameter Extraction

The most frequently seen surfaces of revolution in industry include cylindrical surface. elliptical surface and conic surface. There are several methods of extracting axis parameters from range data [4, 11, 16, 22]. Among them, the one from [11] is that most suitable for our application. Although their method only describes the ways of extracting the parameters of cylindrical surface, more complex surfaces can be analyzed in the same way. In this method, only two scans on the cylindrical surface are needed to extract axis parameters. See Fig. 5, pick two points on one scan, $a_i (i = 1, 2)$; the points should be widely separated on the scan. The method will try to select two points $b_i (i = 1, 2)$ on the second scan so that the dot product of the two unit vectors $v_{i,i}, (i = 1, 2)$ from a_i pointing to b_i is close to 1. Once the points b_i have been found, the unit vectors formed by these two pairs of points $((a_1, b_1), (a_2, b_2))$ are the representation of the direction vector of the axis. Once the axis is known, the estimate of the axis displacement can be computed

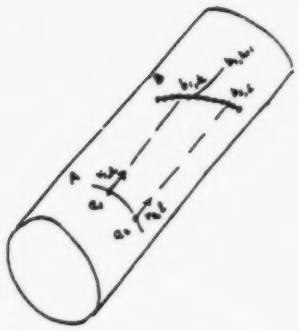

Figure 5: The extraction of axis from two scans ([11])

easily. The projection method used in last section can also be applied here. The difference is that instead of projecting a line-segment on the projecting plane, an arc will be projected on the plane and the parameters of the arc including the radius and the center of the arc will be calculated from the projected data. The two dimensional coordinates of the center wii then be back-projected in the original coordinate system and used as the displacement of the axis.

3.3 Location Determination

Suppose two non-parallel line-segments, boundary edges or axes, have been extracted, $\tilde{\mathbf{l}}_i$ $i = 1, 2$, and are expressed as

$$\tilde{\mathbf{l}}_i = \tilde{\mathbf{l}}_{0_i} + t\tilde{\mathbf{n}}_i \quad i = 1, 2 \quad -\infty < t < +\infty \tag{1}$$

where $\tilde{\mathbf{l}}_{0_i}$ are the vectors from the origin of the sensing coordinate system perpendicular to and intersecting the lines the line-segments lie on, and $\tilde{\mathbf{n}}_i$ are the unit direction vectors of the line-segments viewed from the sensing coordinate system. Their corresponding line-segments $\mathbf{l}_i$ in object model are expressed as

$$\mathbf{l}_i = \mathbf{l}_{0_i} + t\mathbf{n}_i \quad i = 1, 2 \quad -\infty < t < +\infty \tag{2}$$

where $\mathbf{l}_{0_i}$ are the vectors from the origin of the object coordinate system perpendicular to and intersecting the lines the line-segments lie on, and $\mathbf{n}_i$ are the unit direction vector of the line-segments. The location determination is to find a transformation matrix $\mathbf{T}$ such that

$$\tilde{\mathbf{l}}_i = \mathbf{T}\mathbf{l}_i \quad i = 1, 2$$

3.3.1 Determining the Rotation

The rotation can be represented in several ways. If it is represented as a rotation axis $\boldsymbol{\rho}$ of magnitude 1 and a rotation angle α, the rotation can be easily derived by using the

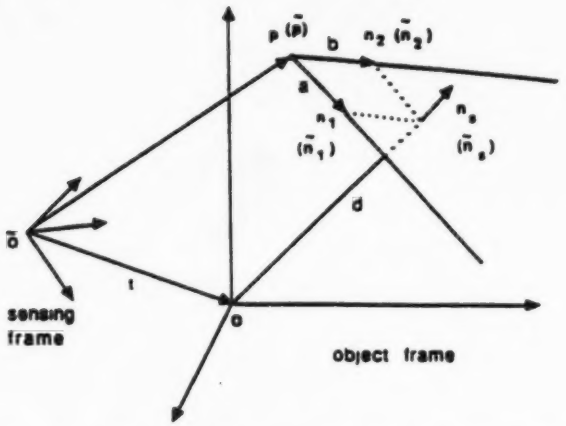

Figure 6: Two line-segments are in the same plane

fact that the rotation axis should be perpendicular to both ($\mathbf{n}_1$-$\tilde{\mathbf{n}}_1$) and ($\mathbf{n}_2$-$\tilde{\mathbf{n}}_2$). In fact, given any two pairs of vectors, the rotation parameters can always be derived. The $\boldsymbol{\rho}$ has the form

$$\boldsymbol{\rho} = \frac{(\mathbf{n}_1 - \tilde{\mathbf{n}}_1) \times (\mathbf{n}_2 - \tilde{\mathbf{n}}_2)}{\|(\mathbf{n}_1 - \tilde{\mathbf{n}}_1) \times (\mathbf{n}_2 - \tilde{\mathbf{n}}_2)\|}$$

to within an ambiguity of degree of π. Once we have computed a direction vector $\boldsymbol{\rho}$, the rotation angle can be derived by using the following formula:

$$\cos\alpha = \frac{\mathbf{n}_i \cdot \tilde{\mathbf{n}}_i - (\boldsymbol{\rho} \cdot \tilde{\mathbf{n}}_i)^2}{1 - (\boldsymbol{\rho} - \tilde{\mathbf{n}}_i)^2}$$

Given values for $\boldsymbol{\rho}$ and α, the rotation part of the transformation matrix can be determined.

3.3.2 Determining the Translation

When considering the determination of the displacement, we have to deal with two possible cases: e.g., the two line-segments are located in the same plane and they are in different planes.

In the first case, see Fig 6, the intrinsic geometric properties of the line-segments are used to derive the formulas. From the given conditions, the intersection point $\mathbf{p}$ of the two lines emitted from the line-segments, the plane normal vector $\mathbf{n}_s$, the distance d of the origin from the plane, all expressed in the object coordinate system, can be calculated. Also from the equation

$$\mathbf{p} + a\mathbf{n}_1 + b\mathbf{n}_2 - d\mathbf{n}_s = \mathbf{o}$$

and using the properties of vector algebra, we can solve for the parameters a and b:

$$\begin{aligned} a &= ((d\mathbf{n}_s - \mathbf{p}) - b\mathbf{n}_2) \cdot \mathbf{n}_1 \\ b &= \frac{(d\mathbf{n}_s - \mathbf{p}) \cdot \mathbf{n}_2 - ((d\mathbf{n}_s - \mathbf{p}) \cdot \mathbf{n}_1)(\mathbf{n}_1 \cdot \mathbf{n}_2)}{1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)^2} \end{aligned}$$

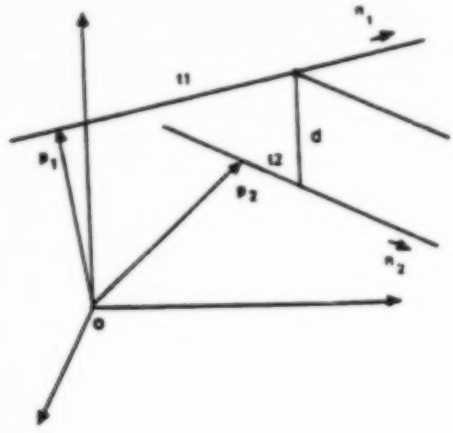

Figure 7: The cross points of two line-segments

Using the constants d, a, b and the extracted line-segment parameters, the translation vector $\mathbf{t}$ is given by

$$\mathbf{t} = \tilde{\mathbf{p}} + a\tilde{\mathbf{n}}_1 + b\tilde{\mathbf{n}}_2 - d\tilde{\mathbf{n}},$$

where $\tilde{\mathbf{p}}$, $\tilde{\mathbf{n}}$, are the intersection point and plane normal expressed in sensing coordinate system, and can be computed from the the extracted line-segment. The correctness of the equation (3.3.2) is based on the fact that during any transformation, the relative position and orientation of all vectors are unchanged.

For the second case, e.g., two line-segments are not in the same plane, we can first find the two cross points of a line which is perpendicular to both of the two line-segments and has the shortest distance. See Fig 7. Once the cross points have been found, all the computations used in case one can be applied to derive the displacement vector.

The coordinates of cross points can be derived by computing the corresponding t values of each of the line-segment equations (1) and (2):

$$t_1 = \frac{(\mathbf{l}_{0_1} - \mathbf{l}_{0_2}) \cdot \mathbf{n}_1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)(\mathbf{l}_{0_1} - lo2) \cdot \mathbf{n}_2}{(\mathbf{n}_1 \cdot \mathbf{n}_2)^2 - 1} \tag{3}$$

and

$$t_2 = \frac{-(\mathbf{l}_{0_1} - lo2) \cdot \mathbf{n}_2 + (\mathbf{n}_1 \cdot \mathbf{n}_2)(\mathbf{l}_{0_1} - \mathbf{l}_{0_2}) \cdot \mathbf{n}_1}{(\mathbf{n}_1 \cdot \mathbf{n}_2)^2 - 1} \tag{4}$$

Because the two line-segments are non-parallel, $(\mathbf{n}_1 \cdot \mathbf{n}_2)^2 - 1$ will not be equal to zero and therefore we do not need to consider the overflow problem here.

4 Using Multi-Line Features to Locate Objects

As we have mentioned, two line segments are enough to completely specify transformation parameters. Sometimes, multi-line parameters can be extracted. These multiple measurements may be used to improve the accuracy of the measurements. In this section,

we present an efficient algorithm which is based on the use of dual number quaternions [24]. The method solves for the orientation and position of an object by minimizing a single cost function associated with the sum of the orientation and position errors. In the following, quaternions are represented with a boldface script character, such as $\boldsymbol{q}$; vectors are represented with a boldface Roman character, such as $\mathbf{q}$; and dual quantities are represented with hat over a character, such as $\hat{\boldsymbol{q}}$.

Quaternions have long been used as a method to represent orientation [13, 19]. The extension of quaternions to include the representation of position and orientation is made by simply changing all of the quantities in the quaternion to dual quantities. Furthermore, the dual quaternion has a similar interpretation as the real quaternion:

$$\hat{\boldsymbol{q}} = \begin{bmatrix} sin(\hat{\theta}/2)\hat{\mathbf{n}} \\ cos(\hat{\theta}/2) \end{bmatrix} \tag{5}$$

where the vector $\hat{\mathbf{n}}$ is now a dual number quaternion representing the unit vector about which the coordinate system has rotated and translated and $\hat{\theta}$ is the dual number quaternion representing the angle of rotation and distance of translation, as illustrated in Fig. 8. The dual vector $\hat{\mathbf{n}}$ and dual angle $\hat{\theta}$ are

$$\begin{aligned} \hat{\mathbf{n}} &= \mathbf{n} + \epsilon \mathbf{p} \times \mathbf{n} \\ \hat{\theta} &= \theta + \epsilon d \end{aligned}$$

where $\mathbf{n}$ is the axis of rotation and $\mathbf{p}$ is the position vector of any point along the line of rotation and translation, θ is the angle of rotation and d is the distance of translation along the line.

The dual quaternion can also be represented in another form:

$$\hat{\boldsymbol{q}} = \boldsymbol{r} + \epsilon \boldsymbol{s} \tag{6}$$

where $\boldsymbol{r}$ and $\boldsymbol{s}$ are both real quaternions and are called the real and dual parts, respectively. From equation 5 and 6, we have the following relationships:

$$\boldsymbol{r} = \begin{bmatrix} \sin(\theta/2)\mathbf{n} \\ \cos(\theta/2) \end{bmatrix} \tag{7}$$

and

$$\boldsymbol{s} = \begin{bmatrix} d/2\cos(\theta/2)\mathbf{n} + \sin(\theta/2)\mathbf{p} \times \mathbf{n} \\ -d/2\sin(\theta/2) \end{bmatrix} \tag{8}$$

Two useful matrix functions of quaternions are the matrices $\boldsymbol{Q}(\boldsymbol{r})$ and $\boldsymbol{W}(\boldsymbol{r})$ which are defined as:

$$\boldsymbol{Q}(\boldsymbol{r}) = \begin{bmatrix} r_4 \boldsymbol{I} + \boldsymbol{K}(\mathbf{r}) & \mathbf{r} \\ -\mathbf{r}^T & r_4 \end{bmatrix} \tag{9}$$

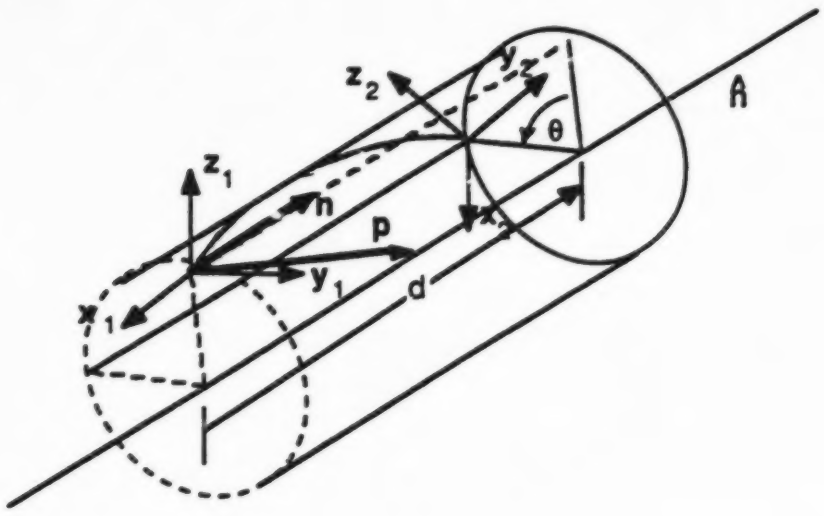

Figure 8: Illustration of rotation and translation for dual quaternion ([24])

$$\boldsymbol{W}(\mathbf{r}) = \begin{bmatrix} r_4\boldsymbol{I} - \boldsymbol{K}(\mathbf{r}) & \mathbf{r} \\ -\mathbf{r}^T & r_4 \end{bmatrix} \tag{10}$$

where $\boldsymbol{K}(\mathbf{r})$ is the skew-symmetric matrix

$$\boldsymbol{K}(\mathbf{r}) = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix} \tag{11}$$

Important properties of these quaternion matrices can be found in [24].

In our application, a line segment is characterized by the pair (n,p), where n is a unit direction vector of the line segment and p is the distance vector to the origin. We also define the position quaternion $\boldsymbol{p}$ and direction quaternion $\boldsymbol{n}$ as the following:

$$\begin{aligned} \boldsymbol{p} &= \frac{1}{2}\begin{bmatrix} \mathbf{p} \\ 0 \end{bmatrix} \\ \boldsymbol{n} &= \begin{bmatrix} \mathbf{n} \\ 0 \end{bmatrix} \end{aligned}$$

Now, suppose m line-segments have been extracted and represented in quaternion form $(\tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{n}}_i)$; and $(\boldsymbol{p}_i^o, \boldsymbol{n}_i^o)$ are the corresponding quaternion representation of modeled line-segments, where $m > 2$ and $i = 1, \cdots, m$. If the estimate of transformation is represented as $\hat{\boldsymbol{q}} = \boldsymbol{r} + \epsilon\boldsymbol{s}$, the estimated line-segment parameters $(\boldsymbol{p}_i, \boldsymbol{n}_i)$ will be

$$\boldsymbol{p}_i = \boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{s} + \boldsymbol{W}(\boldsymbol{r})^T\boldsymbol{Q}(\boldsymbol{r})\boldsymbol{p}_i^o \tag{12}$$

and

$$\boldsymbol{n}_i = \boldsymbol{W}(\boldsymbol{r})^T \boldsymbol{Q}(\boldsymbol{r}) \boldsymbol{n}_i^o \tag{13}$$

The approach for computing the position and orientation of the object is to determine $\boldsymbol{r}$ and $\boldsymbol{s}$ which minimizes the error between the $\tilde{\boldsymbol{p}}_i$ and $\boldsymbol{p}_i$ and the $\tilde{\boldsymbol{n}}_i$ and $\boldsymbol{n}_i$. That is, we select the $\boldsymbol{r}$ and $\boldsymbol{s}$ to minimize the following error function:

$$E = \sum_{i=1}^{m} (\alpha_i (\boldsymbol{n}_i - \tilde{\boldsymbol{n}}_i)^2 + \beta_i (\boldsymbol{p}_i - \tilde{\boldsymbol{p}}_i)^2) \tag{14}$$

where α_i's and β_i's are weighting factors.

Using the properties of dual number quaternions, we can rewrite the error function into the following form

$$E = \boldsymbol{r}^T \boldsymbol{C}_1 \boldsymbol{r} + \boldsymbol{s}^T \boldsymbol{C}_2 \boldsymbol{s} + \boldsymbol{s}^T \boldsymbol{C}_3 \boldsymbol{r} + constant \tag{15}$$

where

$$\boldsymbol{C}_1 = \sum_{i=1}^{m} (-2\alpha_i \boldsymbol{Q}(\tilde{\boldsymbol{n}}_i)^T \boldsymbol{W}(\boldsymbol{n}_i^o) - 2\beta_i \boldsymbol{Q}(\tilde{\boldsymbol{p}}_i)^T \boldsymbol{W}(\boldsymbol{p}_i^o)) \tag{16}$$

$$\boldsymbol{C}_2 = (\sum_{i=1}^{m} \beta_i) \boldsymbol{I} \tag{17}$$

$$\boldsymbol{C}_3 = 2 \sum_{i=1}^{m} \beta_i (\boldsymbol{W}(\boldsymbol{p}_i^o) - \boldsymbol{Q}(\tilde{\boldsymbol{p}}_i)) \tag{18}$$

After considerable matrix algebra, the equation (15) may be written as:

$$E = constant - \lambda \tag{19}$$

where λ is the eigenvalue of the matrix $\boldsymbol{R}$:

$$\boldsymbol{R} = \frac{1}{2} (\boldsymbol{C}_3^T (\boldsymbol{C}_2 + \boldsymbol{C}_2^T)^{-1} \boldsymbol{C}_3 - \boldsymbol{C}_1 - \boldsymbol{C}_1^T) \tag{20}$$

and $\boldsymbol{r}$ is the corresponding eigenvector. Thus, the error is minimized if we select the eigenvector $\boldsymbol{r}$ corresponding to the largest positive eigenvalue.

Having computed $\boldsymbol{r}$, the $\boldsymbol{s}$ can be obtained by using the equation:

$$\boldsymbol{s} = -(\boldsymbol{C}_2 + \boldsymbol{C}_2^T)^{-1} \boldsymbol{C}_3 \boldsymbol{r} \tag{21}$$

The corresponding 3×3 rotation matrix $\boldsymbol{A}$ can be written in terms of the components of the quaternions in the following way:

$$\begin{bmatrix} \boldsymbol{A} & 0 \\ 0^T & 1 \end{bmatrix} = \boldsymbol{W}(\boldsymbol{r})^T \boldsymbol{Q}(\boldsymbol{r}) \tag{22}$$

and the position vector can be written in the form

$$\boldsymbol{p} = \boldsymbol{W}(\boldsymbol{r})^T \boldsymbol{s} \tag{23}$$

In a summary, the method first computes matrix $\boldsymbol{R}$, then finds the maximum eigenvalue and its corresponding eigenvector of the matrix, and finally derives the transformation parameters from it. Among these steps, the computation of matrix $\boldsymbol{R}$ spends most of the time. The complexity of the algorithm is $\mathbf{O}(m)$.

5 Discussion

We have presented our object localization algorithm based on line segment matching. The method is very simple and computationally fast. In most cases, closed-form formulas are used to derive the solution. The method is also quite flexible, because only few surfaces (one or two) need to be accessed (sensed) to gather necessary range data. For example, if the line-segments are extracted from boundaries of a planar surface, only parameters of one surface and two of its boundaries need to be extracted, as compared with traditional point-surface matching or line-surface matching algorithms which need to access at least three surfaces in order to locate a planar object. Therefore, this method is especially suitable for applications when an object is surrounded by many other work pieces and most of the object is very difficult, if not impossible, to be measured; or when not all parts of the object can be reached. The theoretical ground on how to use line range sensor to locate object has been laid on this paper. Much work has to be done in order for the method to be really useful. Currently we are doing the following things:

1. Error analysis including theoretical analysis and simulation result.
2. Calibration of the sensor.
3. Experiments to use the sensor to take measurements on objects with different shapes and materials.

Publications

1. Lejun Shao, Michael W. Walker, and Richard A. Volz, "Estimating 3-D location parameters using dual number quaternion", submitted to *Computer Vision, Graphics, and Image Processing* for publication, May 1989.
2. Lejun Shao, Richard A. Volz and Michael W. Walker, "3-D object location determination using line-segments matching", *Int'l Workshop on Industrial Applications of Machine Intelligence and Vision*", pp. 145-150, April 10 -14, 1989, Tokyo, Japan.

3. M. W. Walker, Lejun Shao and R. A. Volz, "Optimal object localization using dual number quaternions", *SPIE Conference on Applications of Artificial Intelligence. VII*, pp. 603-613, March 28-30, 1989, Orlando, Florida.

4. Lejun Shao, and Richard A. Volz, "Methods and Strategies in Object Localization", *NASA Conference on Space Telerobotics*, January 31-February 2, 1989, Pasadena, California.

References

[1] R. Benton and D. Waters, " *Intelligent Task Automation Interim Technical Reports*", AFWAL/MLTC Wright Patterson AFB, Ohio, Reps. 1-7 Oct. 1985.

[2] P. Horaud, and R. B. Booles, "*3DPO's Strategy for Matching three-Dimensional Objects in Range Data*", Proc. IEEE 1984 Int'l Conf. Robotics, Atlanta, GA, March, 1984, pp. 78-85.

[3] R. B. Booles, P. Horaud, and M. J. Hannah, "*3DPO: Three Dimensional Part Orientation System*", Robotics Research: The First Symposium, Cambridge, MA: MIT Press 1984.

[4] Fred L. Boolstein, " *Fitting Conic Sections to Scattered Data*", CGIP, Vol.9, 1979, pp.56-71.

[5] CyberOptics Corporation, "*Point/Line Range Sensor Instruction Manual,*", 1988.

[6] Faugeras, O.D. and Herbert, M., "*The Representation, Recognition, and, Locating of 3-D Objects*", The Int'l Journal of Robotics Research, Fall 1986, pp.27-52.

[7] John V. Draper, Joseph N. Herndon, and Wendy E. Moore, *The Implications of Force Reflection for Teleoperation in Space*", Goddard Conf. on Space Applications of Artificial Intelligence and Robotics, May 1987.

[8] Kristjan T.Gunnarsson, "*Optimal Part Localization by Database Matching with Sparse and Dense Data*", Ph.D. dissertation, Dept. of Mechanical Engineering, Carnegie Mellon Univ., May 1987.

[9] Kristjan T. Gunnarsson and Friedrich B. Prinz, "*CAD Model-Based Localization of Parts in Manufacturing*", Computer, August 1987, pp.66-74.

[10] W.Eric L. Grimson and Tomas Lozano-Perez, "*Model-Based Recognition and Localization from Sparse Range or Tactile Data*", The Int'l Journal of Robotics Research, Fall 1984 pp.3-35.

[11] Tomas Lozano-Perez, W.Eric L.Grimson, and S.J. White, "*Finding Cylinders in Range Data*", IEEE Int'l Conf. on Robotics and Automation, March 1987, pp.202-207.

[12] Steven J. Gordon and Warren P. Seering, "*Real-Time Part Position Sensing*", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI 10, No.3 May 1988.

[13] Wahba Grace, "*A least Squares Estimate of Satellite Attitude*", problem 65.1, SIAM Review, pp.384-386, July 1966.

[14] Robert E. Keil, Timothy A. Skunes, and Steven K. Case, "*Achieving High Accuracies with Triangulation-Based Range Sensors*", SME Conf. Proceedings on Robot 12 and Vision 88, June 1988, pp.5-107 - 5-120.

[15] S. Linnainmaa, D. Harwood, and L.S. Davis, "*Pose Determination of a Three-Dimensional Object Using Triangle Pairs*", CAR-TR-95, Center for Automation Research, University of Maryland, 1985.

[16] Joon Hee Han, *"Range Image Analysis for 3-D Object Recognition"*, Ph. D. dissertation, EECS Dept., University of Michigan, June 1988.

[17] R. A. Jarvis *"A Perspective on Range Finding Techniques for Computer Vision"* IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No.2, March 1983, pp. 122-139.

[18] Jamas S.Albus, Harry G. Mccain, and Ronald Lumia, *"NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)"*, National Bureau of Standards, Robot System Division, March 13, 1987, pp.12.

[19] Edward Pervin and Jon A. Webb, *"Quaternions in Computer Vision and Robotics,"* Tech. Rep., Dept. of Computer Science, Carnegie-Mellon University, CMU-CS-82-150, 1982.

[20] *"Robotic Assessment Test Sets: Level 1, 2, & 3"*, NASA Gorddard, SS-GSFC-0029.

[21] W.S. Rutkowski, R. Benton, and E.W. Kent, *"Model-Driven Determination of Object Pose for a Visually Served Robot"*, IEEE Int'l Conf. Robotics and Automation, Raleigh, NC, 1987, pp.1419-1428.

[22] Paul D. Sampson, *"Fitting Conic Sections to "Very Scattered" Data: A Iterative Refinement of the Bookstein Algorithm"*, CGIP, Vol. 18, 1982, pp.97-108.

[23] G. Stockman, S. Kopstein, and S. Benett, *"Matching Images to Models for Registration and Object Detection via Clustering"*, IEEE Trans. on Pattern Analysis and Machine Intelligence, PAMI-4, No. 3, 1982, pp. 229-241.

[24] M.W. Walker, L. Shao and R. Volz *"Optimal Object Localization Using Dual Number Quaternions"*, SPE Int'l Conf. on Applications of Artificial Intelligence, March 1989.

[25] D. E. Whitney *Historical Perspective and State of the Art in Robot Force Control"*, IEEE Conf. on Robotics and Automation, 1985.

[26] R.A. Volz, Rajiv Desai, and Jing Xiao, *Contact Formations, A New Approach to Automatic Robot Program Generation*, invited paper at 1988 NATO Workshop on CAD and Robotics, July 4-7, 1988, Italy.

[27] Jing Xiao , and R.A. Volz *Design and Motion Constraints of Part-Mating Planning in the Presence of Uncertainties"*, IEEE Conf. on Robotics and Automation, April 1988.